

<https://helda.helsinki.fi>

Dissemination of GNSS RTK using MQTT

Rao, Ashwin

IEEE
2021

Rao , A , Kirkko-Jaakkola , M & Ruotsalainen , L 2021 , Dissemination of GNSS RTK using MQTT . in 2021 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit) . European Conference on Networks and Communications , IEEE , pp. 282-287 , Joint European Conference on Networks and Communications and 6G Summit , Porto , Portugal , 08/06/2021 . <https://doi.org/10.1109/EuCNC/6GSummit51104.2021.9482565>

<http://hdl.handle.net/10138/333497>

<https://doi.org/10.1109/EuCNC/6GSummit51104.2021.9482565>

submittedVersion

Downloaded from Helda, University of Helsinki institutional repository.

This is an electronic reprint of the original article.

This reprint may differ from the original in pagination and typographic detail.

Please cite the original version.

Dissemination of GNSS RTK using MQTT

*Ashwin Rao, [†]Martti Kirkko-Jaakkola, and *Laura Ruotsalainen

*University of Helsinki, [†]National Land Survey of Finland

Abstract—Precise positioning using Global Navigation Satellite System (GNSS) requires the GNSS receivers to compensate for the errors caused by distortion in the GNSS signal’s path due to atmospheric conditions. The Real Time Kinematics (RTK) technique uses terrestrial reference stations that continuously monitor the quality of GNSS signals and provide information that can be used by the GNSS receivers in the vicinity of a reference station to compensate for the errors.

In this paper, we explore the performance of disseminating the RTK correction information using the Message Queuing Telemetry Transport (MQTT) protocol over 5G. We also compare the indirection costs (latency overheads) of using MQTT over 5G to Ethernet and Wi-Fi, our baselines for high-speed and wireless connectivity respectively, and we highlight the impact of 5G power savings when disseminating GNSS RTK using MQTT.

Index Terms—5G, FinnRef, GNSS, Indirection, MQTT, Power savings, RTK.

I. INTRODUCTION

Precise positioning is vital for a variety of use cases including automated driving, intelligent transportation system, and tracking of industrial devices. These use cases typically rely on the Global Navigation Satellite System (GNSS) for positioning. GNSS receivers compute their position and time from the GNSS signals received from multiple GNSS satellites that transmit positioning and timing data [1].

The signals transmitted by the GNSS satellites traverse through the atmosphere to reach to the GNSS receivers. The Ionosphere and Troposphere are known to distort the signal’s path, and the signal can also be attenuated or reflected by objects such as trees or buildings. Furthermore, the estimated satellite orbits and differences in time have imperfections (see § II). Consequently, consumer grade GNSS receivers have an accuracy of approximately five meters, and this accuracy can be in the order of hundred meters in the events of atmospheric interference. An accuracy of roughly five metres is sufficient for use cases such as geotagging of photographs, however many professional applications such as land surveying, construction, and autonomous driving demand an accuracy on a scale of a few centimetres [2].

One widely used approach to address these errors is to use terrestrial reference stations that continuously monitor the quality of GNSS signals and produce real-time data that can be used to compensate for the errors. Specifically, Real Time Kinematics (RTK) is a technique which provides high positioning performance by using correction information from a base station, that is a few kilometers away from the GNSS receiver (also called a rover) [1].

In this paper, we build on the recommendations of Ruddick *et al.* [3] and present our approach to disseminate RTK cor-

rection information via Message Queuing Telemetry Transport (MQTT) messages (§ III). We explore the performance of our solution when using 5G links, and also compare the indirection costs (latency overheads) of using MQTT over 5G to Ethernet and Wi-Fi, our baselines for high-speed and wireless connectivity respectively (§ IV). Specifically, we observe that power-savings can have significant impact on the latency and jitter, and this can be mitigated using simple approaches such as having a background ping when using 5G.

II. BACKGROUND AND MOTIVATION

A. GNSS primer

Global Navigation Satellite Systems (GNSS) provide global absolute position, velocity and time solution (PVT). The positioning accuracy of current off-the-shelf consumer grade GNSS receivers is about 2-5 m in open-sky environments when using conventional code based positioning [1]. However, these levels of accuracy are insufficient for use cases such as autonomous driving applications which require centimeter-level positioning accuracy [4]. Achieving this level of accuracy requires correcting the errors that degrade the signal quality.

GNSS satellites orbit in space at the distance of around 20000 km from the Earth. While travelling to the Earth, the GNSS signals traverse the Ionosphere, which is the zone of terrestrial atmosphere extending itself from 60 km from the Earth’s surface. The Ionosphere contains free electrons that delay the signal and cause the largest portion of the GNSS receiver errors. This delay depends on the signal frequency, and it may be cancelled out to some extent when using multi-frequency signals. Consequently, all GNSS systems and their regional counterparts—BeiDou, Galileo, GLONASS, GPS, NavIC, and QZSS—transmit signals on multiple frequencies [5]. However, using signals on multiple frequencies improves the accuracy to the order of only one meter. After passing the Ionosphere, the GNSS signals traverse through a layer called Troposphere, resulting again in extra delay in the signal. The magnitude of the delay depends on existing local weather conditions, namely the temperature, air pressure, and humidity. Although models exist for correcting the error caused by the troposphere [6], the improvement in the accuracy remains far from the desired centimeter-level positioning accuracy.

GNSS receivers compute the carrier phase which is a measure of the range between a satellite and receiver expressed in units of cycles of the carrier frequency [7]. However, the resulting distances are not only related to the distance between the receiver antenna and the satellites, but also to an imperfect alignment of the receiver’s time scale to the GPS

time scale called pseudoranges [8]. The carrier phases (ϕ) and pseudorange codes (ρ) are modeled as

$$\rho = \|\mathbf{p} - \mathbf{p}_i - \delta\mathbf{p}_i\| + c(\delta t - \delta t_i) + I_i + T_i + \epsilon_i \quad (1)$$

$$\phi = \|\mathbf{p} - \mathbf{p}_i - \delta\mathbf{p}_i\| + c(\delta t - \delta t_i) - I_i + T_i + \lambda N_i + \eta_i \quad (2)$$

where \mathbf{p} and \mathbf{p}_i denote the positions of the user and the i th satellite, respectively; $\delta\mathbf{p}_i$ denotes the error in the broadcast satellite orbit information used for computing its position at the signal transmission time; the receiver and satellite clock biases are denoted by δt and δt_i , respectively; c is the speed of light and λ denotes the signal wavelength; propagation delays due to the ionosphere and troposphere are represented by I_i and T_i , respectively; and N_i is the carrier phase cycle ambiguity.

The error sources such as measurement noise, multipath, and antenna imperfections are denoted by ϵ_i and η_i for the pseudorange and the carrier phase respectively. In general, the variance of pseudorange noise is larger than its carrier phase counterpart by several orders of magnitude [9]. However, unmodeled errors are receiver's location and type specific, and these errors cannot be corrected using the methods discussed in this paper.

In the following, we briefly discuss the main error sources addressed in our paper, namely Ionospheric and Tropospheric delays, satellite position, and clock errors. The GNSS signals transmitted by satellites contain information that can be decoded to resolve the satellite positions \mathbf{p}_i as well as for correcting the clock error δt_i . A GNSS receiver may estimate its three-dimensional position after data from at least four satellites is successfully decoded. In spite of all the sophisticated means for correcting the errors that deteriorate the signals, the error in the stand-alone position solution in open-sky environments is rarely under one meter [10].

B. Real Time Kinematics (RTK) Primer

Real Time Kinematics (RTK) is capable of providing centimeter level position accuracy by using correction information from a terrestrial base station that is at a known position [1]. This base station periodically send its location, and its code and carrier measurements for all satellites whose signals it is able to receive, to the GNSS receivers that are connected to it. This enables the GNSS receivers to leverage double differencing, i.e., take advantage of the spatial and temporal correlation of ranging errors due to the space segment and the propagation medium [11].

RTK uses the carrier phase measurements and also the correction information for cancelling out the previously discussed errors. By differencing the measurements made for satellite i between the user and the base station, satellite orbit, clock and atmosphere related errors are entirely canceled out or at least significantly mitigated. Furthermore, the measurements are differenced between satellites i and j , which cancels out errors due to the receiver clocks. The maximum distance to the base station depends on the various factors including the ionospheric activity, the number of frequencies supported by rover and base receivers, etc; this maximum distance is around 20 kilometers or less for good performance [12]. Furthermore,

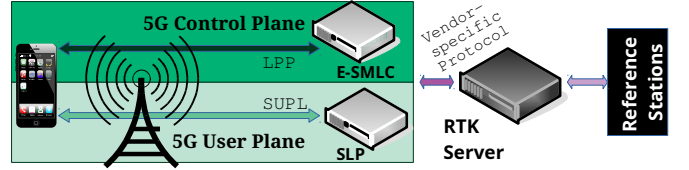


Fig. 1. **Dissemination of RTK in 5G networks.** The reference stations provide the correction information to an RTK server which in turn use vendor specific protocols to disseminate it to the 5G network functions.

this technique requires a communication channel between the terrestrial base stations and the GNSS receivers.

The RTK correction information can be disseminated using the Radio Technical Commission for Maritime Services (RTCM) messages [13] or even vendor specific messages such as the Compact Measurement Record (CMR) [14]. Traditionally, the RTK corrections have been obtained in the form of reference receiver measurements either by deploying a base station or by subscribing to a commercial RTK network. In the rest of the paper, we focus our attention on RTCM messages because they are used by Finnish national GNSS network. Their RTK servers aggregate the correction information from base stations spread across Finland, and they use the Networked Transport of RTCM via Internet Protocol (NTRIP) [15] for disseminating the RTK correction information as RTCM messages.

C. RTK in 5G Non Standalone (NSA) Mode

As shown in Fig. 1, the Evolved Serving Mobile Location Centre (E-SMLC) can use vendor specific protocols to fetch the RTK correction information from the RTK servers. This network function is also responsible for the delivery of this data to the 5G user equipment (UE) that are authorized to receive this information [16]. The UE use the LTE Positioning Protocol (LPP) and the Secure User Plane Location (SUPL) to exchange location information with the E-SMLC in the control-plane and the SUPL Location Platform (SLP) in the user plane respectively.

As discussed in § III, the availability of the RTK correction information when using 5G will be limited to the users of the mobile network network. Furthermore, this service will be available only in the coverage area of the mobile network.

III. DISSEMINATION OF CORRECTION INFORMATION

The dissemination of the correction information can be performed using various approaches, including the following.

(1) *Each client independently asks the RTK server for correction information.* A key issue with this approach is that it will require the RTK server to scale with the number of clients. The RTK servers typically use NTRIP for dissemination of RTK correction information as RTCM messages. Ruddick *et al.* [3], show that servers using NTRIP do not scale with the number of clients; they argue for using MQTT to disseminate the RTCM information.

(2) *The location server of the communication network collects and disseminates the correction information for geographical coverage area of network.* This is the approach

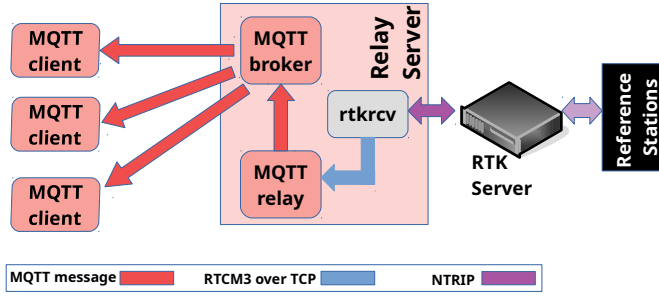


Fig. 2. **Dissemination of RTK using MQTT.** The *rtkrcv* binary of *RTKlib* receives information from the RTK servers which is then disseminated to the clients using the MQTT protocol. The MQTT client is running on the end-user devices that require precise positioning.

leveraged by the 5G. As shown in Fig. 1, the 5G network will contact the RTK server for the correction information and disseminates it to its clients. While this approach can scale with the number of clients, its availability is limited to the users and coverage area of the mobile network.

(3) *The RTK server publishes the correction information to a server that disseminates this information to the clients.* This approach builds on the strengths of previous two approaches. Specifically, like approach (1) it allows each client to independently request for RTK correction information regardless of the network to which it is connected, and like approach (2) it minimizes the load on the RTK server and the GNSS infrastructure collecting the information from the terrestrial reference stations. In Fig. 2, we present the key components of our implementation of this approach.

For our prototype we used the RTK server of the Finnish national GNSS network (FinnRef) as the source for GNSS correction data [17]. FinnRef is a network of almost 50 continuously operating GNSS reference station located in Finland. It represents a cost-effective alternative between a commercial network and global service providers such as the International GNSS Service [18]. Furthermore, some FinnRef stations also contribute to the International GNSS Service. The FinnRef RTK server encodes the RTK correction information as RTCM messages and allows clients to request the RTK correction information using the NTRIP.

As shown in Fig. 2, our Relay Server uses NTRIP to receive data from the FinnRef RTK server. Specifically, we use the *RTKRCV* application of *RTKLIB* [19] to request information via RTCM messages. *RTKRCV* enables exporting the data received from the RTK server in various formats including exporting the information on a TCP connection. We configure the *RTKRCV* instance on our Relay Server to export the received RTCM messages over a TCP connection. At the other end of this TCP connection is our Message Queuing Telemetry Transport (MQTT) relay application. This application publishes the received RTCM3 messages to an MQTT broker running on the Relay Server. Specifically it encapsulates the received RTCM3 messages inside MQTT messages and publishes the MQTT messages to the MQTT broker. This MQTT broker can accept subscription requests for RTCM information from clients.

TABLE I
SOFTWARE USED FOR EVALUATION

Software	Version	Notes
RTKLIB	2.4.3-b33	<i>rtkrcv</i> configured with input <i>ntripcli</i> and output of <i>tcpvsvr</i>
Mosquitto	1.6.9	MQTT Broker on Relay
Paho-MQTT	1.5.0	Python library used by the MQTT publisher on the Relay and the MQTT subscriber on the Client
Ubuntu	20.04	OS on the Relay and the Client
OxygenOS (Android)	10.0.10	5G Enabled OnePlus 7 Pro Phone

In our prototype, the MQTT clients running on user devices subscribe to the RTCM3 information from the MQTT broker; the clients currently do not publish any information to the MQTT broker, and they simply subscribe to the RTCM information. Furthermore, our Relay Server—built using *RTKRCV*, a MQTT Relay, and an MQTT Broker—decouples the RTK server from its clients. Consequently, the RTK server has no knowledge on the number of clients subscribing to the RTCM messages published at the broker, and this isolates the RTK Server from their load. This approach is inspired by the work done by Ruddick *et al.* [3] who advocate the benefits of using MQTT over NTRIP for publishing RTCM3 information. Note that the MQTT messages received by the client contain the original RTCM messages exported by the *RTKRCV* application running on our Relay Server.

The RTCM information received by *RTKRCV* is specific to given a geographical location. In our prototype, the *RTKRCV* subscribes to the correction information for the location at which our client is running. This can be easily extended by including multiple *RTKRCV* instances, where each *RTKRCV* instance will request the correction information for a given geographical area. The number of *RTKRCV* instances will depend on the coverage area of the RTK server. Each *RTKRCV* instance will communicate with a dedicated MQTT relay that will publish the RTCM messages along with the coverage area for which the messages are intended to. The clients can then subscribe to the RTCM messages for their geographical area. We detail this approach in § V, and in the following we present an evaluation of the performance of our current prototype.

IV. EVALUATION

A. Goal

The goal of our evaluation is to quantify the delays incurred by the indirection introduced by our system. Specifically, the delays incurred when clients use our approach over 5G to request the correction information from the RTK server.

B. Setup

1. Relay Server.

We host our Relay Server on a computer with Intel(R) Xeon(R) CPU E5540 and 48 GB of RAM. As shown in Tab. I, our Relay Server runs the Mosquitto MQTT broker [20] and the *RTKRCV* client of *RTKLIB* [19]. We use the Paho-MQTT python library for our MQTT relay application that receives RTCM messages from *RTKRCV*, and encapsulates

and publishes the received message within MQTT messages. Along with publishing the RTCM3 messages, the MQTT relay also publishes the timestamp at which the messages were received. We use this timestamp with the time at which the MQTT message was received by the clients to measure the delays in receiving the RTCM messages. We use the Network Time Protocol (NTP) [21] to synchronize the clock of our Relay Server.

2. MQTT Client.

We use the Paho-MQTT python library for our MQTT client that subscribes to the RTCM messages arriving at the MQTT broker. As mentioned previously, the RTCM messages are encoded in the received MQTT messages that also include the timestamp at which the MQTT relay received the encoded RTCM message from RTKRCV. The client logs this timestamp along with the time at which it received the MQTT messages; our client also uses NTP to synchronize its clock. For our evaluation we used an HP Elitebook 840 G3 laptop with an Intel i5-6300 CPU and 16 GB of RAM running Ubuntu 20.04. We also use a 5G enabled One Plus 7 Pro phone and use USB tethering to allow the client to use 5G to communicate with our Relay Server. We compare the delays incurred when using 5G at a given location with the delays incurred when using Ethernet and Wi-Fi at the same location.

3. Accounting for Power Saving Optimizations.

Wireless devices, including Wi-Fi devices on our laptops and the 5G modems on our phones, perform power saving optimizations. Hoque *et al.* [22] highlight the impact of these optimization on network traffic measurements and recommend steps to disable them, and also steps to work around them—such as ensuring the battery level is above a particular level. To quantify the impact of Wi-Fi power savings, we conduct our evaluation with Wi-Fi power saving enabled, henceforth denoted as *Wi-Fi (PE)*, and with Wi-Fi power savings disabled, denoted as *Wi-Fi (PD)*. Furthermore, when Wi-Fi power savings were disabled we also ran a background ping [23] from our client laptop to our Relay Server; the interval between successive ping packets was the round-trip time (RTT). This stream of traffic ensures that the radio is not put in an idle state without overloading the network. Similarly, to quantify the impact of 5G power savings, we run our evaluation in two settings: (a) where we tried to disable power savings by running a background ping from our client laptop to our Relay Server, denoted as *5G (PD)*, and (b) when this ping was not running resulting in the default power savings being enabled, denoted as *5G (PE)*.

C. Evaluation Locations

We conduct our experiments at two different locations which we categorize as University and Residence. At each location, we evaluate the performance of our system using Ethernet, Wi-Fi, and 5G; Ethernet and Wi-Fi act as our baselines for high-speed and wireless connectivity respectively, and as shown in Fig. 2, our implementation does not use any 5G network functions for receiving the RTK correction information. Al-

though these two locations cannot be used to generalize all possible geographical locations, they provide insights on the performance one can expect when using our solution.

The University location represents our laboratory at the University of Helsinki. We chose this location because it has an indoor 5G Base Station operated by a telecom operator in Finland. This base station provides us with a sample point that can be used to represent geographical locations with excellent coverage. The NetRadar application [24] showed that the 5G signal strength was consistently stronger than -85 dBm at the location at which we conducted our experiments. Similarly, the Wi-Fi signal strength was stronger than -60 dBm at this location. Our Relay Server was also hosted on a computer at this location.

The Residence location represents the residence of one of our authors, and it offered Wi-Fi and Ethernet connectivity using a DSL connection from the same telecom operator. The signal strength of 5G signals was consistently weaker than -90 dBm at this location. This location therefore provides us with a sample point that can be used to represent geographical locations with poor cellular coverage.

D. Results

1. Inter-Arrival time of RTCM Messages.

In Fig. 3, we present the distribution of the inter-arrival time between consecutive NTRIP packets from the RTKRCV application. We use this figure to highlight the time budget for disseminating the RTCM messages.

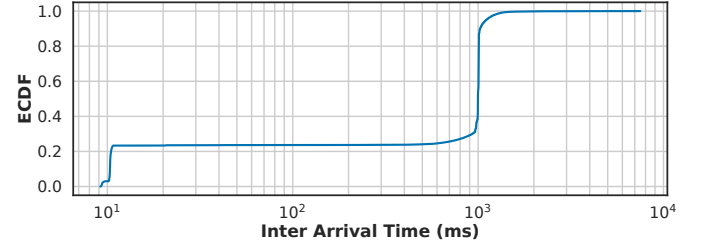


Fig. 3. Distribution of the Inter-arrival time of NTRIP packets that contain the RTCM3 messages from the Finnish GNSS RTK server. RTCM3 messages can be batched together and sent in the same NTRIP packet. We observe that the RTCM3 messages arrive typically once every second.

The RTCM3 messages currently arrive at a frequency of 1 Hz or a multiple of 1 Hz from the FinRef RTK server. This arrival rate is representative of the typical message transmission frequency of RTCM messages from RTK servers. Each NTRIP packet can contain multiple RTCM3 messages, and this batching of RTCM3 messages implies that the RTK server sends multiple NTRIP packets in succession if a batch of RTCM3 messages cannot fit in one packet. As a consequence of this batching, we observe that the inter-arrival time of about 20% of the NTRIP packets have an inter-arrival time of less than 10 ms, and close to 70% of the messages have an inter-arrival time which is close to 1 second.

2. Network Latency Between the Relay Server and Client.

We first ran ping for 1800 seconds from our MQTT Client to our Relay Server for each communication technology from

each location. We use these measurements to quantify the latency between our client and server for each combination of the location and communication technology. These observations act as a baseline for the desired latency when using a given communication technology at a given location.

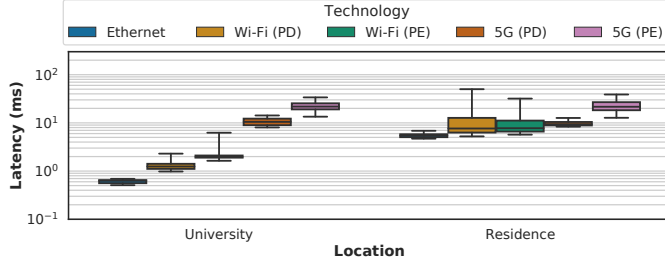


Fig. 4. **Network latency measured using ping.** The latency shown in this figure is half of the reported round trip time (RTT), i.e. $RTT/2$. The error bars denote the 5th and the 95th percentile of the observed values.

In Fig. 4, we present the network latency between our MQTT client and our Relay Server for different communication technologies in our two locations. The latency shown in this figure is half of the RTT reported by ping, and the error bars of the box plot denote the 5th and the 95th percentile of the observations. The latency for Ethernet and Wi-Fi from the Residence is larger than that at the University because our Relay Server was located in the University network. We also observe the impact of the background ping on the 5G latency. Specifically, the latency is significantly higher when there is no background traffic because of the power savings that increase the delay. The time interval between successive ping packets is 1 second, and this interval is large enough to trigger the power savings and put the radios in an idle state. Waking up the radio results in an additional delay, and as demonstrated by Da Silva *et al.* [25], this trade off between power savings and latency can be addressed by power saving configurations at the UE and also the radio access network (RAN).

2. Indirection Costs.

We measure the indirection costs by running the MQTT client and Relay Server for 1800 seconds for each communication technology and from each location. The objective of these measurements was to quantify the latency and jitter when relaying the RTCM3 messages via our Relay Server.

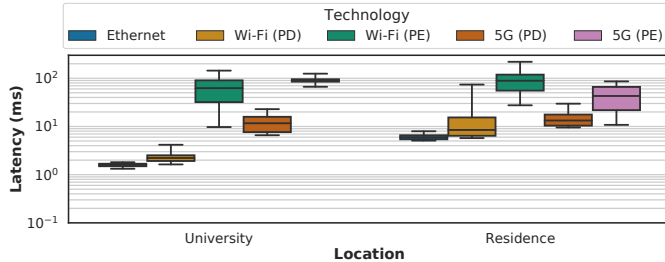


Fig. 5. **Delays caused by indirection.** The box plot present time taken by the MQTT message to arrive at our client from our MQTT relay. The error bars denote the 5th and the 95th percentile of the observed values.

In Fig. 5, we present the the latency incurred by the indirection. At the University, as expected, the latency for

Ethernet is the smallest. At both locations, we also observe that the range of the latency incurred when Wi-Fi power savings are enabled are significantly larger when they are disabled. We observe a similar behavior when there is background traffic over 5G links. Specifically, the latency over 5G when power savings are disabled is significantly lower than the latency incurred over Wi-Fi when power savings are enabled.

Along with the latency, we also measure the difference in the jitter when using a communication technology at a given location. We compute the jitter difference as follows. We measure the inter-arrival of time between the k^{th} packet and the $k+1^{th}$ packet at the Relay as $\delta t_{r,k} = t_{r,k} - t_{r,k+1}$ where $t_{r,k}$ and $t_{r,k+1}$ denote the timestamps at which the k^{th} and $k+1^{th}$ packet arrived at the Relay. Similarly, we measure the corresponding inter-arrival time of these packets at the MQTT client as $\delta t_{c,k} = t_{c,k} - t_{c,k+1}$. We use these two values to measure δt_k , the difference in the jitter at the Relay Server and MQTT client; $\delta t_k = |\delta t_{r,k} - \delta t_{c,k}|$. Clearly, it is desirable to have a small jitter difference to ensure that the experience is agnostic to the location and communication technology.

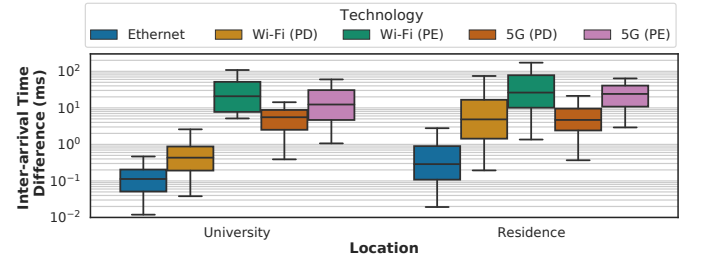


Fig. 6. **Jitter Difference.** A small difference in the jitter is desirable.

In Fig. 6, present the our observations on the jitter difference. As expected the jitter difference for Ethernet and Wi-Fi is more at the Residence compare to the University. Furthermore, the we do not observe a significant difference in the jitter for 5G networks when background traffic is enabled, which is also expected because mobile networks are designed for location agnostic service quality.

In summary, we observe that MQTT can be used to disseminate the RTK correction information, and the overheads of using MQTT (Fig. 5, and Fig. 6) are comparable to the latency increase caused by indirection (Fig. 4).

V. DISCUSSION

1. Scaling to span a given geographical area.

Although the RTCM information received by RTKRCV is specific to given geographical location, our approach can be easily scaled by including multiple RTKRCV instances where each RTKRCV instance will request the correction information for a given geographical area. Each RTKRCV instances can publish a location-specific correction information via a MQTT relay that is dedicated to this instances. The MQTT broker decouples the MQTT clients from the RTKRCV instances, and allows the client to subscribe to the RTCM messages for their current geographical area. This can be achieved in various ways. For instance, the MQTT broker can advertise

separate topics for each geographical area, and the client can then decide the topics for it would like to subscribe to. This approach allows clients to pro-actively subscribe to correction information of geographical areas based on its trajectory. Alternatively, the client can provide its location to the MQTT broker, and the broker can ensure that the client receives the stream of RTCM information that is relevant to its location. The strengths and weakness of each of these approaches requires further evaluation.

2. Secure Dissemination of RTK.

The RTKRCV instance running on our Relay Server receives the correction information using NTRIP. During our evaluation we observed that NTRIP does not timestamp the RTCM messages, making the data exchange vulnerable to replay attacks. This vulnerability can be mitigated using approaches for ensuring *at-most-once* message delivery [26]. Our Relay Server therefore timestamps the messages sent. Furthermore MQTT supports payload encryption that can protect the RTCM3 messages from tampering.

VI. CONCLUDING REMARKS

The RTK technique is capable of providing centimeter level position accuracy by using correction information from terrestrial base stations. In this paper, we explore the performance of disseminating the RTK correction information using the MQTT protocol over 5G, and compare it to using Wi-Fi and Ethernet links. We observe that the overheads of using MQTT are comparable to the latency increase caused by indirection.

ACKNOWLEDGEMENTS AND AVAILABILITY

The code used for the experiments and results used to plot the figures are publicly available at <https://version.helsinki.fi/ashwin.rao/5give>. This work has been carried out in the ESA AO-1-9161 5GIVE project, funded by the European Space Agency (ESA) and the University of Helsinki. The 5G device and the 5G connectivity was provided by the Business Finland 5G FORCE project.

REFERENCES

- [1] E. Kaplan *et al.*, *Understanding GPS Principles and Applications*. Norwood, MA, USA: Artech House, 2006.
- [2] “Study on positioning use cases,” 3GPP, Tech. Rep. 3GPP TR 22.872, v.16.1.0, Sep. 2018.
- [3] R. Ruddick *et al.*, “Improving the efficiency of GNSS data streaming within the IGS,” *International GNSS Service Workshop*, 2018.
- [4] H. Wymeersch *et al.*, “5g mmwave positioning for vehicular networks,” *IEEE Wireless Communications*, vol. 24, no. 6, pp. 80–86, 2017.
- [5] E. S. A. (ESA), *Navipedia*. [Online]. Available: https://gssc.esa.int/navipedia/index.php/Main_Page, (accessed: 14.2.2020).
- [6] J. Saastamoinen, “Contributions to the theory of atmospheric refraction,” *Part II Refraction Corrections in Satellite Geodesy. Bulletin Geodesiques*, pp. 13–34, 1973.

- [7] M. Petovello *et al.*, “Carrier phase and its measurement for gnss,” *GNSS Solutions*, 2010.
- [8] M. Rao *et al.*, “Code tracking and pseudoranges,” *Inside GNSS*, vol. 7, pp. 26–33, 2012.
- [9] P. D. Groves, *Principles of GNSS, Inertial, and Multi-sensor Integrated Navigation Systems*. ProQuest Ebook Central, Artech House, 2013.
- [10] F. Peyret *et al.*, *SaPPART white paper: Better use of global navigation satellite systems for safer and greener transport*, ser. Les collections de l’Ifsttar techniques et methodes, TMI 1. French Institute of Science, Technology for Transport, Development, and Networks, 2015.
- [11] M. Kirkko-Jaakkola *et al.*, “Effect of antenna location on gnss positioning for its applications,” in *European Navigation Conference (ENC)*, 6. 2016, vol. 24, pp. 1–7.
- [12] *Real Time Kinematics - Navipedia*. [Online]. Available: https://gssc.esa.int/navipedia/index.php/Real_Time_Kinematics, (accessed: 14.2.2020).
- [13] “Differential GNSS (Global Navigation Satellite Systems) Services,” Radio Technical Commission for Maritime Services (RTCM), Tech. Rep. RTCM 10403.1.
- [14] “CMRx: A New Correction Format From Trimble,” Trimble, Tech. Rep. CMRx.
- [15] “Standard for Networked Transport of RTCM via Internet Protocol (Ntrip),” Radio Technical Commission for Maritime Services (RTCM), Tech. Rep. RTCM 10410.1.
- [16] “LTE Positioning Protocol,” 3GPP, Tech. Rep. 3GPP TS 37.355 version 15.0.0 Release 15, Jan. 2020.
- [17] *FinnRef GNSS stations*. [Online]. Available: <https://www.maanmittauslaitos.fi/en/research/research/other-research-and-measuring-stations/finnref-gnss-stations>, (accessed: 14.2.2020).
- [18] *International GNSS Service*. [Online]. Available: <https://www.igs.org/network>, (accessed: 14.2.2020).
- [19] *RTKLIB: An Open Source Program Package for GNSS Positioning*. [Online]. Available: <http://www.rtklib.com/>, (accessed: 14.2.2020).
- [20] *Eclipse Mosquitto - An open source MQTT broker*. [Online]. Available: <http://mosquitto.org/>, (accessed: 14.2.2020).
- [21] *NTP - The Network Time Protocol*. [Online]. Available: <http://www.ntp.org/>, (accessed: 14.2.2020).
- [22] M. A. Hoque *et al.*, “Network and Application Performance Measurement Challenges on Android Devices,” *ACM SIGMETRICS PER*, vol. 48, no. 3, pp. 6–11, 2021.
- [23] *IP Utils*. [Online]. Available: <https://github.com/iputils/iputils>, (accessed: 14.2.2020).
- [24] *Netradar App*. [Online]. Available: <https://www.netradar.com/netradar-app/>, (accessed: 14.2.2020).
- [25] I. L. Da Silva *et al.*, “A novel state model for 5G Radio Access Networks,” in *IEEE ICC Workshops*, 2016, pp. 632–637.
- [26] B. Liskov *et al.*, “Efficient at-most-once messages based on synchronized clocks,” *ACM Transactions on Computing Systems*, vol. 9, no. 2, pp. 125–142, May 1991.